ИЗВЕСТИЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ

2006 МАТЕМАТИКА № 4 (527)

КРАТКИЕ СООБЩЕНИЯ

УДК 519.852

А.С. ВЕЛИЧКО

ОБ АЛГОРИТМЕ ДВОЙСТВЕННЫХ ОТСЕЧЕНИЙ ДЛЯ ЗАДАЧИ ДВУХЭТАПНОГО СТОХАСТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Введение

Оптимизационные задачи стохастического программирования [1] появляются при разработке математических моделей многих экономических или технических систем. Это связано с тем, что либо параметры таких моделей заранее точно не известны, что более характерно для экономических систем, либо проектируемое изделие предназначено для работы в случайных и не полностью предсказуемых условиях, что более типично для технических устройств.

Среди возможных постановок наибольшее распространение получила линейная двухэтапная модель перспективного решения и последующей коррекции, которая состоит в минимизации средних потерь с учетом последующей оптимальной модификации [2], [3]. Эти задачи имеют характерные структурные особенности в матрицах ограничений, учитывающие неантисипативный характер коррекций и независимость альтернативных сценариев. Такие особенности предоставляют определенные возможности для декомпозиции структурированных задач на относительно слабо связанные блоки меньшей размерности и использования параллельных алгоритмов.

В качестве основной модели структурированной оптимизационной задачи рассматривается двублочная задача линейного программирования со связывающими переменными [4], [5], для ее решения используется прямо-двойственный алгоритм декомпозиции, основанный на двойственных отсечениях [6], [7].

1. Постановка структурированной линейной оптимизационной задачи

В качестве основной модели структурированной оптимизационной задачи рассмотрим двублочную проблему следующего вида:

$$\min_{z_A, z_B, x} c_A z_A + c_B z_B, \tag{1}$$

$$A_A z_A + B_A x \le d_A, \tag{2}$$

$$A_B z_B + B_B x \le d_B, \tag{3}$$

$$z_A \ge 0, \quad z_B \ge 0, \tag{4}$$

где z_A, z_B — векторы переменных задачи, x — вектор связывающих переменных, A_A, A_B, B_A, B_B — матрицы соответствующих размерностей.

При фиксированном x эта задача распадается на два независимых блока, что используется для развития декомпозиционного алгоритма двойственных отсечений.

Введя функции

$$f_A(x) = \min_{z_A \ge 0} c_A z_A, \quad f_B(x) = \min_{z_B \ge 0} c_B z_B, \tag{5}$$
$$A_A z_A \le d_A - B_A x, \quad A_B z_B \le d_B - B_B x,$$

получим эквивалентную (1)–(4) задачу

$$\min_x\{f_A(x) + f_B(x)\}. \tag{6}$$

Используя сопряженные функции

$$h_A(p) = \max_x\{px - f_A(x)\}, \quad \text{и} \quad h_B(p) = \max_x\{px - f_B(x)\},$$

задачу (6) перепишем в терминах сопряженных функций [6]

$$\min_p\{h_A(-p) + h_B(p)\}. \tag{7}$$

Такая эквивалентность задач (6) и (7) позволяет организовать эффективный процесс обмена координирующей информацией между прямыми и двойственными задачами линейного программирования в декомпозиционном подходе к решению задачи (1)–(4).

2. Задача двушагового стохастического программирования

В линейном случае возникает следующая математическая постановка:

$$\min c(\omega)x, \tag{8}$$

$$S(\omega)x = h(\omega), \quad x \geq 0, \tag{9}$$

где ω — элементарное событие на заданном вероятностном пространстве (Ω, F, Π).

Решение $x(\omega)$ будет зависеть от реализации определенного события ω. Однако, если решение этой задачи должно быть определено *a priori*, вряд ли найдется вектор x, удовлетворяющий ограничениям задачи для всех событий $\omega \in \Omega$. Вектор $\delta(x,\omega) = h(\omega) - S(\omega)x$ характеризует степень нарушения ограничений исходной модели при выборе какого-то x.

В двушаговой модели стохастического программирования предполагается, что на первом шаге известны лишь возможные реализации параметров модели, на втором шаге получается полная информация о наступившем событии ω. В таком случае оптимальный выбор x может проводиться при одновременной минимизации некоторого функционала от вектора $\delta(x,\omega)$. Выбирая в качестве такого функционала сумму абсолютных значений компонент вектора, запишем задачу в виде

$$\min_{x\geq 0}[Ec(\omega)x + \min E\{ey^+ + ey^-,\ y^+ - y^- = h(\omega) - S(\omega)x, y^+ \geq 0,\ y^- \geq 0\}],$$

где E — символ математического ожидания.

В случае, когда все множество событий Ω содержит всего два события ω_1, ω_2, вероятности наступления которых равны соответственно p_1, $p_2 = 1 - p_1$, последняя задача имеет вид

$$\min\{(p_1c^1 + p_2c^2)x + p_1(ey_1^+ + ey_1^-) + p_2(ey_2^+ + ey_2^-)\},$$

$$y_1^+ - y_1^- + S_1x = h_1, \quad y_2^+ - y_2^- + S_2x = h_2, \quad x, y_1^+, y_1^-, y_2^+, y_2^- \geq 0,$$

где e — вектор из единиц.

Ясно, что система ограничений данной задачи является блочной вида (1)–(4). Однако вектор связывающих переменных x в данной задаче входит в минимизируемый функционал.

Обозначим новый вектор связывающих переменных через $\widetilde{x}$. Добавим ограничение $x = \widetilde{x}$ и агрегируем его, например, с ограничениями $y_1^+ - y_1^- + S_1x = h_1$. Дальнейшее агрегирование вектора x с векторами y_1^+, y_1^- позволяет исключить вхождение вектора x в функционал получаемой задачи, сохраняя двублочность ограничений.

В результате такой агрегации получим эквивалентную для (1)–(4) структурированную двублочную задачу

$$\begin{gathered}\min\{\widetilde{c}_1\widetilde{y}_1 + \widetilde{c}_2\widetilde{y}_2\},\\ A_A\widetilde{y}_1 + B_A\widetilde{x} = \widetilde{h}_1,\\ A_B\widetilde{y}_2 + S_2\widetilde{x} = h_2,\\ \widetilde{x}, \widetilde{y}_1, \widetilde{y}_2 \geq 0,\end{gathered}$$

где $\widetilde{y}_1 = (x,\ y_1^+,\ y_1^-)$, $\widetilde{y}_2 = (y_2^+,\ y_2^-)$, $\widetilde{c}_1 = (p_1c^1 + p_2c^2,\ p_1e,\ p_1e)$, $\widetilde{c}_2 = (p_2e,\ p_2e)$, $A_A = \left(\begin{smallmatrix} S_1 & E & -E \\ E & 0 & 0\end{smallmatrix}\right)$, $B_A = \left(\begin{smallmatrix} 0 \\ -E\end{smallmatrix}\right)$, $\widetilde{h}_1 = \left(\begin{smallmatrix} h_1 \\ 0\end{smallmatrix}\right)$, $A_B = \left(\begin{smallmatrix} E & -E\end{smallmatrix}\right)$, E — единичная матрица, e — вектор из единиц.

3. Алгоритм двойственных отсечений

В [7] был предложен декомпозиционный алгоритм двойственных отсечений для решения задач вида (6). Алгоритм требует модификации ([6], с. 1245) в случаях возникновения несобственных подзадач в процессе своего выполнения. Суть используемого алгоритма заключается в замене функции $f_B(x)$ в (6) и $h_A(p)$ в (7) на свои внешние кусочно-линейные аппроксимации $f_B^k(x) = \max\{f_B^{k-1}(x),\ \overline{p}^kx - h_B(\overline{p}^k),\ p^kx - h_B(p^k)\}$ и $h_A^k(p) = \max\{h_A^{k-1}(p),\ p\overline{x}^k - f_A(\overline{x}^k),\ px^{k+1} - f_A(x^{k+1})\}$.

Первый блок содержит последовательное решение задач $\overline{p}^k \in \partial f_B(x^k)$ и $\min\limits_{x\in X_d^{m-1}}\{f_A(x)+f_B^k(x)\}$. Во втором блоке решаются задачи $\overline{x}^k \in \partial h_A(-p^k)$ и $\min\limits_{p\in P_d^{n-1}}\{h_A^k(-p) + h_B(p)\}$. Указанные задачи минимизации являются аппроксимированными вариантами задач (6), (7), и в результате их решения определяются субоптимальные векторы x^k, p^k. Вычисления значений функций $f_B(x)$, $h_A(p)$ и их субградиентов в точках x^k, $-p^k$ выступают подзадачами алгоритма. Решение подзадач на каждом шаге алгоритма позволяет строить новые линейные отсечения, которые уточняют кусочно-линейные аппроксимации $f_B^k(x)$ и $h_A^k(p)$.

Другая группа отсечений аппроксимирует множество определения функций $f_B(x)$ и $h_A(p)$. Множества $X_d^m = X_d^{m-1} \cup \{x : \widetilde{p}^mx \leq \widetilde{g}^m\}$ и $P_d^n = P_d^{n-1} \cup \{p : \widetilde{x}^np \leq 0\}$ обновляются в случае возникновения неограниченных или недопустимых подзадач алгоритма, когда генерируемые алгоритмом векторы x^k, $-p^k$ не принадлежат области определения функций $f_B(x)$ и $h_A(p)$.

Важными вычислительными процедурами, которые сводятся к решению задач линейного программирования, являются задачи нахождения субградиентов функций $f_B(\cdot)$ и $h_A(\cdot)$. Пусть $f_B(x)$ определена соотношением (5). Тогда для некоторого фиксированного вектора $\overline{x}$

$$\begin{gathered}f_B(\overline{x}) = \min_{z_B,x} c_Bz_B,\\ A_Bz_B + B_Bx \leq d_B,\\ x = \overline{x}, \quad z_B \geq 0.\end{gathered}\tag{10}$$

Теорема 1. *Пусть задача* (10) *разрешима для заданного* $\overline{x}$, p^* *является вектором оптимальных двойственных переменных к ограничению* $x = \overline{x}$ *в задаче* (10). *Тогда* $p^* \in \partial f_B(\overline{x})$, *и обратно, любой вектор* $p^* \in \partial f_B(\overline{x})$ *является вектором оптимальных двойственных переменных, соответствующих ограничению* $x = \overline{x}$ *в задаче* (10).

По определению сопряженная для $f_A(\cdot)$ функция представляется следующим образом:

$$\begin{gathered}-h_A(-p) = \min_{z_A,x}(c_Az_A + px),\\ A_Az_A + B_Ax \leq d_A, \quad z_A \geq 0.\end{gathered}\tag{11}$$

Теорема 2. *Если* $h_A(p)$ *— сопряженная к* $f_A(x)$ *функция, то оптимальный вектор* x^*, *получаемый при вычислении* $h_A(-\overline{p})$, *обладает свойством* $x^* \in \partial h_A(-\overline{p})$.

При доказательстве данных теорем используются результаты теории двойственности в линейном программировании. В приведенных выше формулировках $f_B(\cdot)$, $h_A(\cdot)$ являются выпуклыми кусочно-линейными функциями, в этих условиях удается доказать сходимость алгоритма [7].

Для сокращения вычислений при решении задачи (11) симплекс-методом применяется процедура "досчета", когда оптимальный вектор задачи, полученный на k-м шаге алгоритма, используется в качестве допустимого решения этой же задачи на $(k+1)$-м шаге. Такая стратегия допустима при решении задачи (11), поскольку при вычислении $h_A(-p^k)$ и $\overline{x}^k \in \partial h_A(-p^k)$ ограничения задачи не изменяются. Заметим, что в задаче (10) такой досчет осуществить не удается из-за ограничения $x = \overline{x}$, в котором вектор $\overline{x}$ изменяется на каждом шаге алгоритма. Поэтому при прочих равных условиях эта задача может оказаться более трудоемкой по времени выполнения, чем задача (11). Более подробно вычислительные аспекты алгоритма описаны в работе [6].

Вычислительные эксперименты проводились на многопроцессорном вычислительном комплексе МВС-1000/16 в составе центра коллективного пользования "Дальневосточный вычислительный ресурс" [8]. Количество ограничений исходной задачи в вычислительных экспериментах достигало 2000. Алгоритм демонстрирует линейный характер сходимости. В работах [9], [10] проводились численные эксперименты для жестких экстремальных задач с тривиальным носителем и экономической задачи о репликации портфеля рыночных активов. В этих существенно различных по структуре задачах используемый алгоритм показал аналогичную практическую вычислительную сложность.

Литература

1. Юдин Д.Б. *Задачи и методы стохастического программирования.* – М.: Сов. радио, 1979. – 392 с.
2. Birge J.R. *Decomposition and partitioning methods for multistage stochastic linear programs* // Operat. Research. – 1985. – № 33. – P. 989–1007.
3. Fragniere E., Gondzio J., Vial J.-P. *Building and solving large-scale stochastic programs on an affordable distributed computing system* // Annals Operat. Research. – 2000. – № 99. – P. 167–187.
4. Лэсдон Л.С. *Оптимизация больших систем.* – М.: Мир, 1975. – 432 с.
5. Цурков В.И. *Декомпозиция в задачах большой размерности.* – М.: Наука, 1981. – 351 с.
6. Величко А.С., Нурминский Е.А. *Опыт декомпозиции метода конечных элементов с использованием теории структурированных оптимизационных задач* // Электронный журнал "Исследовано в России". – 2002. – С. 1237–1256. `http://zhurnal.ape.relarn.ru/articles/2002/113.pdf`
7. Нурминский Е.А. *Численные методы выпуклой оптимизации.* – М.: Наука, 1991. – 168 с.
8. Центр коллективного пользования "Дальневосточный вычислительный ресурс" ИАПУ ДВО РАН. `http://www.dvo.ru/bbc`
9. Величко А.С., Нурминский Е.А. *Прямо-двойственная декомпозиция для жестких экстремальных задач* // Информационный бюллетень ассоциации математического программирования. – Екатеринбург. – 2003. – № 10. – С. 65–68.
10. Величко А.С., Нурминский Е.А. *Прямо-двойственная декомпозиция задачи о репликации портфеля рыночных активов* // Автоматика и телемеханика. – 2004. – № 2. – С. 170–178.

Дальневосточный государственный университет

Поступила
15.03.2005